GASP: Gaussian Avatars with Synthetic Priors

Jack Saunders* University of Bath
Charlie Hewitt Microsoft
Yanan Jian Microsoft
Marek Kowalski Microsoft
Tadas Baltrušaitis Microsoft

Yiye Chen* Georgia Institute of Technology
Darren Cosker Microsoft and University of Bath
Virginia Estellers Microsoft

Nicholas Gydé Microsoft
Vinay P. Namboodiri University of Bath
Benjamin E. Lundell Microsoft

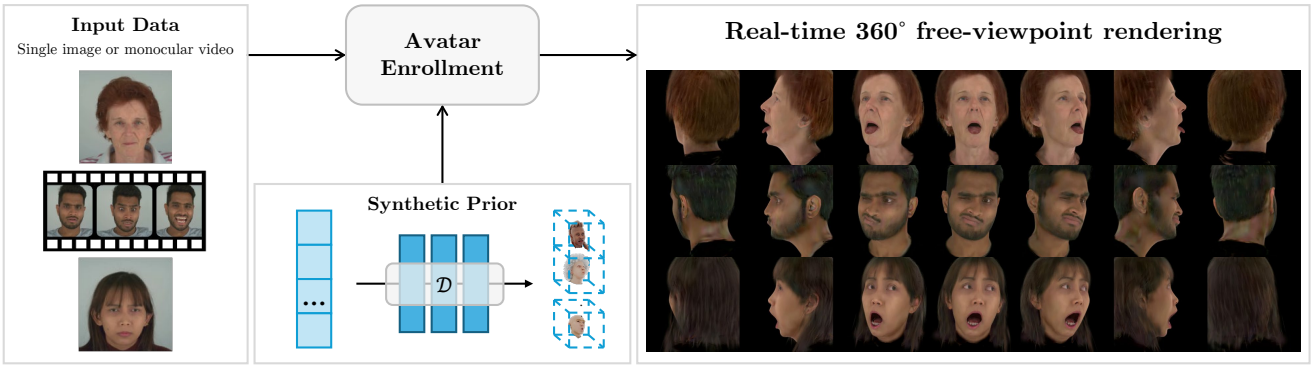

Figure 1. We propose **GASP**, a novel model for creating photorealistic, real-time, animatable, 360° avatars from easily-captured data. We train a generative prior model of Gaussian Avatars on Synthetic data. The prior allows our model to be fit using a single image or a short video with the prior accounting for the unseen views. This lets users create their avatar with only a webcam or smartphone.

Abstract

Gaussian Splatting has changed the game for real-time photo-realistic rendering. One of the most popular applications of Gaussian Splatting is to create animatable avatars, known as Gaussian Avatars. Recent works have pushed the boundaries of quality and rendering efficiency but suffer from two main limitations. Either they require expensive multi-camera rigs to produce avatars with free-viewpoint rendering, or they can be trained with a single camera but only rendered at high quality from this fixed viewpoint. An ideal model would be trained using a short monocular video or image from available hardware, such as a webcam, and rendered from any view. To this end, we propose ***GASP: Gaussian Avatars with Synthetic Priors****. To overcome the limitations of existing datasets, we exploit the pixel-perfect nature of synthetic data to train a Gaussian Avatar prior. By fitting this prior model to a single photo or video and fine-tuning it, we get a high-quality Gaussian Avatar, which supports 360° rendering. Our prior is only required for fitting, not inference, enabling real-time applications. Through our method, we obtain high-quality, animatable Avatars from limited data which can be animated and rendered at 70fps on commercial hardware.*

1. Introduction

Creating high-quality digital humans unlocks significant potential for many applications, including Virtual Reality, gaming, video conferencing, and entertainment. Digital humans must be photorealistic, easy to capture and capable of real-time rendering. The vision and graphics communities have long worked towards this goal, and we are rapidly approaching the point where such digital humans are possible.

A series of works based first on NeRFs [30] raised the

* Work conducted while at Microsoft.

bar in creating exceptional visual quality [4, 5, 9, 14, 23, 54]. However, NeRFs remain slow to render and are unsuitable for real-time applications. Gaussian Splatting-based works have led to significant improvements in both quality and rendering speed [7, 11, 20, 33, 38, 45, 46]. Despite these improvements, the list of suitable applications for these methods is small. Each of these models suffers from one of two drawbacks: either they require expensive capture setups with multiple synchronized cameras, which prevents easy user enrollment [11, 33, 46], or they train on a single camera but exhibit significant quality degradation when rendered from views with more than a minimal variation in camera pose [7, 38, 45]. Furthermore, to maximize visual quality, some of these methods use a large CNN *after* rendering, which prevents real-time rendering without a powerful GPU [11, 46].

For mass adoption, an avatar model should achieve high-quality 360° rendering in real-time and require only the amount of data a user can practically provide. In most cases a user can only capture a monocular, frontal image or video using their webcam or smartphone camera. The problem of fitting an avatar to this data is ill-posed; the extreme sides and back of the head are not visible, leading to artifacts in these unseen regions. In order to overcome this, we require a prior model that is able to "fill in the gaps" left by missing data. Such a model has been shown to be effective in other data-limited, human-centric models, such as visual dubbing [36] and static NeRF models [4]. Ideally, we would train such a model on a large, multi-view, perfectly annotated and diverse dataset. However, very few multi-view face datasets exist. Those that do either lack full coverage, particularly around the back of the head [23], or have only a small number of subjects [44]. Furthermore, annotations such as camera calibrations and 3D morphable model (3DMM) parameters associated with these datasets have to be estimated using imperfect methods and are a significant source of error.

We propose **GASP: Gaussian Avatars with Synthetic Priors**. We use a large, diverse dataset of *synthetic* humans [16, 43] to overcome the difficulties associated with training a prior on real data. This data is generated using computer graphics and has perfectly accurate annotations, including exact correspondence to the underlying 3DMM. This enables the large-scale training of a prior for Gaussian Avatars for the first time. However, the use of synthetic data introduces a domain gap problem. We address this by learning per-Gaussian features with semantic correlations. By learning these correlations on synthetic data and then maintaining them when fitting to real data, using a three-stage fitting process, we can cross this domain gap. Our method even enables rendering the back of the head, having fit to only a single front-facing image or video; see Fig. 1.

To summarize, we propose a novel system for creating realistic, real-time animatable avatars from a webcam or smartphone enabled by the following contributions:

- A prior model over Gaussian Avatar parameters trained using purely synthetic data.
- A three-stage fitting process, combined with learned per-Gaussian correlations to overcome the synthetic-to-real domain gap and allow for 360° rendering.
- Real-time rendering enabled through use of neural networks only during training and fitting, and not at inference time.

2. Related Work

2.1. Photorealistic Animatable Avatars

A significant number of works have attempted to build photorealistic 3D Avatars that can be animated. Most of these works use an existing animatable model, known as a 3D morphable model (3DMM) [3, 24]. Earlier works improve the realism of a 3DMM in image space using compositing [40], a CNN model [22] or pixel-level MLPs [28]. Some works [37, 42] improve the CNN models by adding a learnable latent texture known as a neural texture [41] and evaluating this with a deferred neural renderer. Other works make use of volumetric rendering, either in the form of a point-based representation [52], or a NeRF [14, 29, 31, 49, 53–55]. The primary issue with most of these methods is that they are too slow to render [14, 29, 49, 53, 54], or can only be rendered from limited viewpoints [55].

Gaussian Splatting [20] has allowed for unprecedented photorealism and real-time capabilities in volumetric rendering for Avatars. We refer to this class of methods as Gaussian Avatars. Most Gaussian Avatar methods have built upon 3DMMs as a coarse representation of the geometry and Gaussian Splatting for finer geometry and appearance. Some explicitly bind the Gaussians to the 3D mesh [33, 45], while others learn functions to deform the Gaussians based on 3DMM parameters [7, 11, 46]. These methods are much faster than NeRF-based models but are either only produce good results on cameras close to the training view [7, 45] or require multiple cameras [11, 33, 46]

Furthermore, other approaches based on a 3DMM may learn a texture and/or mesh displacements [1, 2, 6, 12, 13] with optional CNNs to improve quality. These methods are often fast to render and have good novel view synthesis, however they fall short of photorealism [2, 12, 13], take too long to train [6] or deal poorly with hair [1].

2.2. Few-Shot Avatars

Several other works have attempted to address a similar problem to ours, in which the goal is to create a photorealistic avatar from limited amounts of data. In each case, the solution is to leverage some form of data-driven prior. Preface [5] uses a large-scale dataset to train an identity-

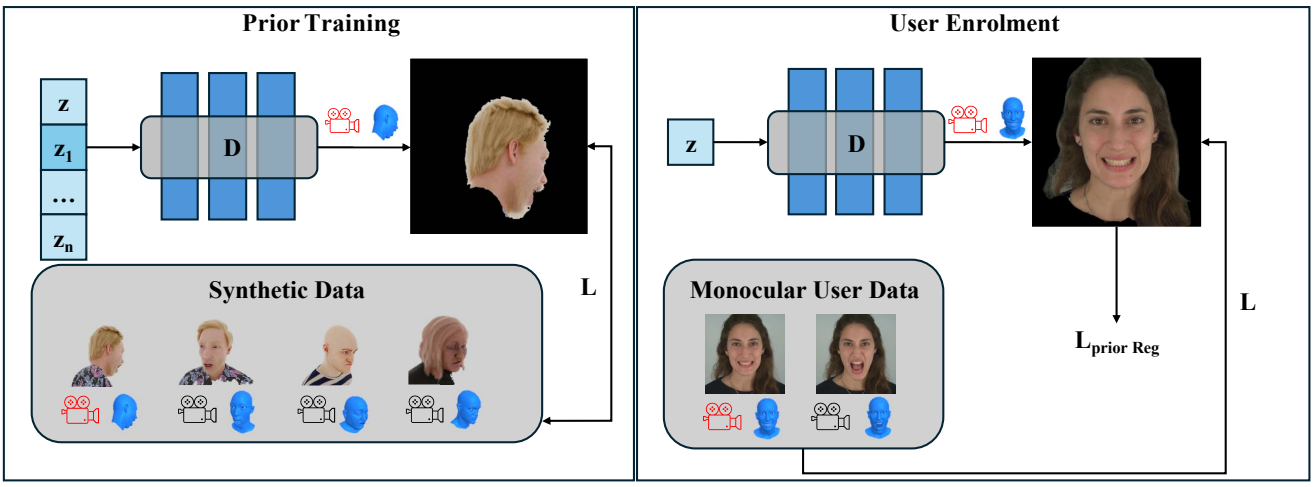

Figure 2. Overview of our method. In the first stage, we train an auto-decoder prior model on Synthetic data to predict the parameters of a mesh-attached Gaussian Avatar. We can then adapt this model to user enrollment data, either a single image or short monocular video. We leverage the prior to improve the quality in unseen regions and achieve free-viewpoint rendering.

conditioned NeRF prior model in an auto-decoder fashion. Cafca [4] also seeks to train this model on large-volume synthetic data. While high quality, the results are static and cannot be animated, and being NeRF-based, are also slow to render, taking over 20s per frame. Some works use powerful 2D image-space models as a prior, exploiting a small amount of data to enable control over the larger model with a 3DMM. StyleRig [39] first achieves control over StyleGAN2 [19] in this way, and DiffusionRig [8] obtains even better results using a DDPM [17] as a prior. Dubbing for Everyone [36] uses a StyleGAN-based UNET with personalized Neural Textures, which allows for better adaptation. ROME [21] takes a similar approach, with neural textures predicted from images. However, as they operate at the image level, they cannot model the back and sides of the head.

Hong et al. [18] build a morphable model for NeRFs using an image dataset for fast and generalisable avatars, however due to its image only training it often displays artefacts from viewpoints outside the narrow range in these datasets. Xu et al. [48]'s work is most similar to ours. They also build a morphable model, this time using Gaussians, which enables fitting to a single image. However, this method does not have a mechanism to update the back of the head and lacks semantic expression parameters, making it harder to animate with existing techniques.

Our work is *real-time*, *animatable* using readily available tools and *photorealistic*. It has a *fast fitting process* which *can be fit using a single camera* and *rendered from any viewpoint, including the back of the head*. While the above works share some of these criteria, none satisfies all. We note that *none* of the above methods have an explicit mechanism for updating the appearance at the back of the head using information from the front, and none *show* the back of the head in their results. A tabular comparison with recent work is provided in the supplementary material.

3. Method

3.1. Background: Gaussian Splatting

3D Gaussian Splatting is a method for reconstructing a volume from a set of images with corresponding camera calibrations. It involves using a collection of Gaussian primitives, represented by a position, $\boldsymbol{\mu}$, in 3D space, an anisotropic covariance matrix, Σ, a color, $\mathbf{c}$ and an opacity, $\boldsymbol{\alpha}$. Kerbl et al. [20] proposed a system to optimize these parameters to fit the evidence provided by the images by decomposing the covariance, Σ, into scale, $\boldsymbol{\sigma}$, and rotation, $\mathbf{r}$, components, represented as a vector and quaternion respectively. Following projection by the camera and depth sorting, each pixel color, P, is computed as:

$$P = \sum_{i=1}^{N_G} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{1}$$

Since the whole process is differentiable, the Gaussian Attributes can be optimized to match the given images and camera parameters.

3.2. Background: Mesh Attached Gaussians

Gaussian Splatting is excellent at reconstructing static scenes but, in its basic form, cannot model animation dynamics. To do this, we make use of a 3D Morphable Model (3DMM) [16, 43] as a way of animating the Gaussians. Multiple works [33, 38] make the observation that, given a sufficiently good coarse approximation of geometry in the

form of a mesh, the problem can be reduced to an approximately static scene. By attaching each Gaussian, $\mathcal{G}_i$, to a specific triangle, $\mathbf{t}$, in the 3DMM mesh, the Gaussian is assumed to remain static relative to that triangle's pose. There are several successful formulations of this posing transformation:

$$\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{r} = \mathcal{T}_{local \rightarrow global}(\boldsymbol{\mu}', \boldsymbol{\sigma}', \mathbf{r}' \mid \mathbf{t}) \tag{2}$$

For our purposes, we use the definition of Qian et al. [33], where the origin of each triangle's system is assumed to be the center, the orthonormal basis is determined by one edge, the triangle's normal and their cross-product, and the isotropic scale by the mean of the length of one edge and its perpendicular in the triangle. This allows us to define a Gaussian Avatar, $\mathcal{G}$, as a collection of static Gaussian primitives in a triangle-local space.

$$\mathcal{G} = \{\mathcal{G}_i : 1 \leq i \leq N_G\},\ \mathcal{G}_i = \{\boldsymbol{\mu}'_i, \boldsymbol{\sigma}'_i, \mathbf{r}'_i, \mathbf{c}_i, \mathbf{o}_i\} \tag{3}$$

As these are static, we can optimize them using the same procedures as in the original formulation [20].

3.3. Prior Model Training

We train our prior model as a generative model over identities. Following previous work [10, 32, 34, 47], we train this prior as an auto-*de*coder model. We jointly learn a per-subject identity code, $\mathbf{z}_j \in \mathbb{R}^{512}, j \in \{1, \ldots, N_{id}\}$, and an MLP decoder, $\mathcal{D}(\mathbf{z})$. One may naïvely think of training this model to directly output the Gaussian Attributes, $\mathcal{A}$, with a single MLP. However, such a method quickly becomes intractable. As a typical model with 100,000 Gaussians may have millions of attributes, the number of parameters in $\mathcal{D}$ would be too large. Instead, we augment each Gaussian with a learnable feature vector, $\mathbf{f}_i \in \mathbb{R}^8, i \in \{1, \ldots, N_G\}$. This feature is analogous to a positional encoding with additional semantic meaning. We then train a network to map these per-Gaussian features to Gaussian attributes, with each Gaussian processed independently and in parallel.

To make optimization more stable, we learn a Canonical Gaussian Template, $\mathcal{C}$, which is fixed across all subjects, and model the per-person variation as offsets from this template. The Canonical Template can be considered the mean Avatar. The i-th Gaussian of the avatar for the subject j is given by:

$$\mathcal{A}_{i,j} = \mathcal{C}_{i,j} + \mathcal{D}(\mathbf{f}_i, \mathbf{z}_j) \tag{4}$$

This is best understood by following Fig. 3. To train this model, we jointly optimize $\mathcal{C}$, $\mathcal{D}$, $\{\mathbf{z}_j\}_{1 \leq j \leq N_{id}}$, $\{\mathbf{f}_i\}_{1 \leq i \leq N_G}$ to minimize the following loss function:

$$\mathcal{L} = \lambda_{pix} L_{pix} + \lambda_{\alpha} L_{\alpha} + \lambda_{percep} L_{percep} + L_{reg} \tag{5}$$

Where L_{pix} is a pixel level loss consisting of L_1, the ℓ_1 difference between the real and predicted images, and L_{SSIM} which is the differentiable SSIM loss, weighted by λ_1 and λ_{SSIM} respectively. L_{percep} is a perceptual loss based on LPIPS [51], L_{α} is the ℓ_1 distance between the real and predicted alpha masks, and L_{reg} is a regularization loss acting on the Gaussians. We regularize scale, σ', and displacement, μ':

$$L_{reg} = \lambda_{\sigma} ||max(0.6, \sigma')||_2 + \lambda_{\mu} ||\mu'||_2 \tag{6}$$

Unlike previous methods, our 3DMM does not capture coarse hair geometry, meaning the Gaussians must fully model it. We, therefore, reduce λ_{μ} by a factor of 100 for Gaussians bound to faces in the scalp region, which we manually define.

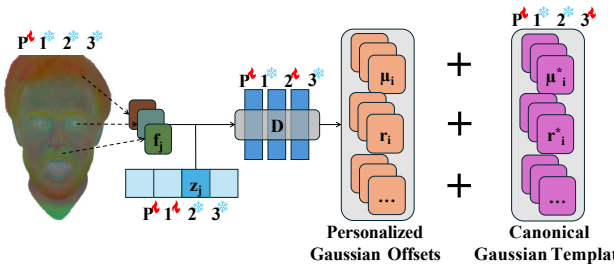

Figure 3. The architecture of our prior model. A latent vector for identity, $\mathbf{z}$, is used to transform learnable per-Gaussian features, $\mathbf{f}$, into Gaussian Attributes, which offset a canonical template. Our training process has four stages: the prior training, P, and three user-specific fitting steps. We freeze some layers and train others at each stage, as indicated.

3.4. Initialization

Using just one Gaussian per triangle face of the 3DMM leads to an under-parameterised model that lacks sufficient detail. To overcome this, we use the initialization strategy of Xiang et al. [45]. We generate a UV map of a given resolution for our mesh and take each pixel's corresponding face and barycentric coordinates. The face is used for Gaussian binding. We use the barycentric coordinates to position the origin of each Gaussian's local coordinate system.

3.5. Fitting Process

Given input data ranging from a single image, to a short video from a single monocular camera, we aim to produce a high-quality avatar that can be viewed from any angle. We have three stages to this fitting process, visualized in Fig. 3:

1. We find the best in-prior Gaussian Avatar by randomly initializing an identity latent vector, $\mathbf{z}$, and optimizing this with everything else frozen; we call this inversion.
2. We fine-tune the MLP, $\mathcal{D}$, with the rest of the model frozen.
3. We refine the resulting Gaussians using the standard Gaussian Splatting optimization procedure [20] to best fit the data.

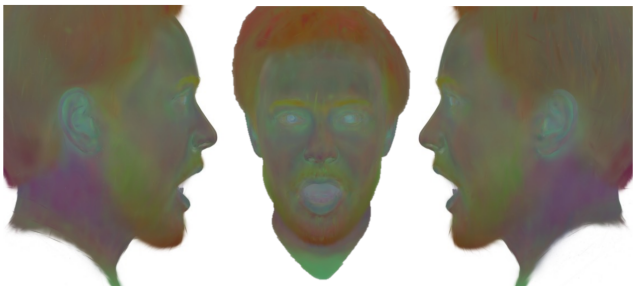

Figure 4. Visualization of the first three components of a PCA decomposition of the Gaussian features $\mathbf{f}$, displayed using the geometry of a random subject. Note the semantic relationships.

Figure 5. Examples from our synthetic dataset. We generate a large and diverse set of synthetic subjects rendered from many views to train our prior model.

To motivate this three-step process, we can consider two extremes. On the one hand, we could perform inversion only. This relies heavily on the prior. If we had perfectly diverse real-human data and a perfect prior, this may be all we would need to do. However, our prior was trained on synthetic data, so inversion can only generate synthetic-looking avatars. On the other hand, we could use the prior for initialization and then optimize the resulting Gaussians. This would achieve similar results to the existing state-of-the-art but with the unseen regions looking synthetic.

We can extract more value from our prior model by considering correlations in the per-Gaussian features, $\mathbf{f}$. Our network is forced to map these to Gaussian attributes and learns to associate similar Gaussians with similar features. Fig. 4 shows a PCA decomposition of the Gaussian features, demonstrating that these features have learned semantic meaning. By freezing the features in the fitting process, Gaussians with similar semantic features will be mapped to have similar attributes. For example, if $\mathcal{D}$ learns to make a Gaussian representing hair at the front of the head blonde, it will also update an unseen one at the back of the head.

To prevent stages 2 and 3 from diverging too far from the prior, we introduce an additional regularization term, L_{prior}, to the loss, $\mathcal{L}$, during these stages. L_{prior} is defined as the ℓ_2 distance between each Gaussian attribute and its corresponding value from the prior (i.e., after stage 1). This is particularly important when regularizing unseen Gaussians. Results after each stage of fitting are shown in the supplementary material.

4. Dataset

We require calibrated multi-camera data of the same subject performing a wide range of expressions to train our prior model. Collecting such data would require complex and expensive camera rigs. Instead, we leverage the synthetic data generation pipeline of Hewitt et al. [16]. This allows us to generate highly diverse and perfectly calibrated image data with pixel-perfect annotations. We generate 1000 identities (random face shape, texture, upper body clothing, hairstyle, hair color, and eye color). We illuminate the scene using uniform white lighting to simplify model training. We pose those faces with random expressions and sample a virtual camera uniformly from a hemisphere ([-180, +180] degrees azimuth and [-20, +45] degrees elevation) to render 50 images per identity. Examples of the data used in training our prior is shown in Fig. 5.

5. Results

We conduct all of our evaluations on the NeRSemble Dataset [23]. This dataset contains multiple subjects performing dozens of facial expression sequences, including one freeform sequence, across 16 cameras. For each sequence, we preprocess each video using an off-the-shelf background removal [25] and face segmentation tool [50] to get the head region only. We obtain Morphable Model parameters in the format of Wood et al. [43] using the method of Hewitt et al. [16]. We consider three experimental settings using this data; please refer to the supplementary material for the cameras and sequences used:

Monocular: To best replicate our desired setting, we enroll all avatars using a single frontal camera. We use a subset of the expression sequences for fitting and evaluate them using the unseen freeform sequence. We use the four most extreme view cameras for evaluation, as determined by manual inspection, to test the model's ability to produce good results on regions unseen at training.

Multi-Camera: To confirm that our model does not sacrifice performance when more data is available, we also enroll avatars using the same configuration above but with all cameras used for input.

Single Image: To test the limits of our model, we also experiment with just a single image as input, selecting the first frame from the Monocular setting as input.

To evaluate visual quality, we use the standard metrics of PSNR, SSIM and LPIPS [51] and FID [15]. We find that PSNR and SSIM prefer solutions that match low-frequency detail, e.g. a flat sheet of hair. While FID is better at cap-

	Monocular Video						Single Image					
Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	ID-SIM ↑	QUAL ↑	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	ID-SIM ↑	QUAL ↑
FlashAvatar	17.25	0.603	0.450	351	0.234	2.08	13.26	0.490	0.519	367	0.057	2.05
GaussianAvatars	17.39	0.601	0.428	366	0.179	2.08	14.80	0.474	0.475	385	0.000	2.03
ROME*	-	-	-	-	-	-	15.78	0.543	0.441	136	0.408	3.38
HeadNeRF*	-	-	-	-	-	-	17.72	0.658	0.333	171	0.114	-
FLARE	14.94	0.524	0.444	140	0.459	-	15.492	0.534	0.440	158	0.322	-
DiffusionRig	19.67	0.343	0.436	155	0.302	2.98	16.87	0.316	0.541	183	0.239	3.15
Ours	21.34	0.712	0.333	117	0.568	3.68	20.73	0.677	0.348	119	0.526	3.80

Table 1. **Quantitative Evaluations:** We compare our method with three state-of-the-art models. We evaluate on two scenarios, for the Monocular scenario we on a single camera and then evaluate on the four most extreme. For single image we do the same but using only the first image from the Monocular sequence. In each case the evaluation sequence is unseen in the training set. We take the average PSNR, SSIM and LPIPS scores for each frame of each avatar. We also ask for user ratings of the quality of each method and report the mean scores out of 5 (QUAL). (*) ROME and HeadNeRF only support single image use cases. We highlight the Best and Second Best for each metric.

Figure 6. Qualitative comparisons of our method with existing state-of-the-art in the **Monocular Setting**. We train on a monocular camera and evaluate on unseen camera poses (top three rows) and an unseen sequence from the training view (bottom row). Our model captures identity better than Diffusion Rig [8] and suffers from fewer artifacts than other Gaussian Avatar models ([33, 45], ours without a prior).

turing high-frequency similarity, we also conducted a user study to measure perceived quality most accurately. We ask users to rate each video out of five and report the mean scores; we denote this QUAL. More details can be found in the supplementary.

5.1. Baselines

We compare our model to state-of-the-art methods. For the first set of methods, we look at Gaussian Avatar models: Gaussian Avatars [33], which is designed for ultra-high quality rendering when trained on multiple views, and Flash Avatars [45], which is designed to be trained and evaluated on monocular data. We train these using the same mor-

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	ID-SIM ↑	QUAL ↑
FlashAvatar	24.73	0.815	0.253	125	0.767	3.70
GaussianAvatars	23.73	0.812	0.285	113	0.773	3.65
DiffusionRig	19.42	0.377	0.425	155	0.302	3.00
Ours	23.44	0.786	0.261	101	0.734	3.80

Table 2. **Multi-Camera:** We run comparisons using 16 training cameras. We report the mean user ratings out of 5 (QUAL). We highlight the Best and Second Best for each metric.

phable model, 3DMM fitting process and dataset preprocessing as our method. In addition to Gaussian Avatar models, we look at models designed for few-shot animatable avatar synthesis. We select the publicly available implementations of ROME [21], DiffusionRig [8] and HeadNeRF [18].

5.2. Monocular Training

The results of this experiment can be found in Tab. 1. Our model significantly outperforms state-of-the-art across all metrics, including user-perceived quality. Our model produces significantly fewer artifacts in novel views compared to other Gaussian Avatar methods [33, 45]. This is because our prior helps prevent the model from overfitting to the training camera view. Diffusion Rig [8] does not show any visible artifacts, but struggles to preserve the identity of the subject, this is best seen in Fig. 6.

5.3. Single Image Training

The results of the single image setting are shown in Tab. 1. With such limited data, other Gaussian Avatar methods overfit and perform poorly. Even on the same camera view as the input image, Gaussian Avatar methods struggle with artifacts; see Fig. 7. Our method also outperforms ROME [21] (on all metrics), and HeadNeRF [18] (on all but one metric) which are designed to work with single images.

5.4. Multi-Camera Training

Our model is competitive with the state-of-the-art in the Multi-Camera setting (Tab. 2). We expect our model to perform worse than other Gaussian Avatar methods [33, 45] as the prior regularizes the model towards a synthetic solution, and we do not model lighting or dynamic expressions. Despite this, our model performs similarly to the state-of-the-art, suggesting it can effectively use all available data. Furthermore, using the prior allows our model to converge in fewer steps than other Gaussian Avatar models, making it cheaper and more efficient to train. Our model performs better on all metrics compared to Diffusion Rig [8].

5.5. Ablations

We perform an ablation study to demonstrate our model's effectiveness. The results are shown in Tab. 3. We use three

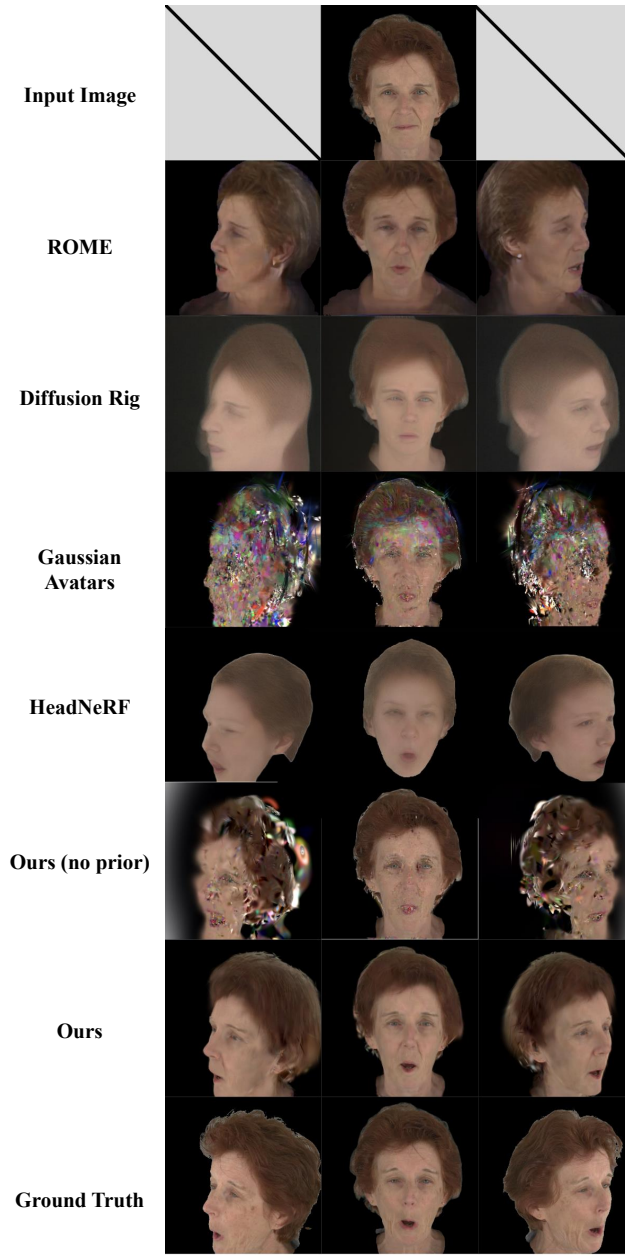

Figure 7. Qualitative comparisons of our method with existing state-of-the-art in the **Single Image Setting**, using the top image only for the fitting process.

subjects in the monocular setting. More details, as well as additional qualitative results, are in the supplementary.

No Prior: To validate the use of the prior, we fit person-specific models using our MLP without any prior. We also ablate the use of the prior regularization loss term. It can be seen that the absence of the prior dramatically reduces the quality according to all metrics, while not regularizing towards the prior leads to slightly better ID reconstruction

Figure 8. Example Avatars from in-the-wild scenarios, including in non-uniform lighting.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	ID-SIM ↑
w/o prior	19.42	0.670	0.391	212	0.478
w/o prior regularization	20.31	0.701	0.344	122	0.620
w/o stage 1	19.56	0.678	0.364	127	0.588
w/o stage 2	20.33	0.704	0.347	118	0.585
w/o stage 3	20.47	0.711	0.343	113	0.441
1 Prior Subject	15.86	0.550	0.459	274	0.365
10 Prior Subjects	19.98	0.678	0.367	146	0.538
100 Prior Subjects	20.39	0.703	0.347	129	0.577
64×64 Gaussians	20.41	0.709	0.363	155	0.493
128×128 Gaussians	20.45	0.704	0.353	127	0.567
256×256 Gaussians	20.43	0.702	0.350	117	0.575
Full (1k Subjects, 512×512)	20.67	0.716	0.340	108	0.589

Table 3. **Ablations:** We ablate several components of the model. We evaluate the absence of the prior, the effect of fewer subjects and fewer Gaussians. We also ablate each stage of the fitting process. We highlight the Best and Second Best for each metric.

but worse quality according to all other metrics.

Number of Subjects: We compare the model quality using priors trained on differing numbers of subjects. The more subjects we have, the better the quality. Interestingly, using one subject in the prior performs worse than not using a prior. This contrasts with the findings of Buehler et al. [4].

Number of Gaussians: We consider the effect of initializing with fewer Gaussians. We use texture maps ranging from 64×64 up to the full 512×512. We see that the highest resolution model performs best. Although the gain is small, it is notable visually (see supplementary).

Fitting Stages: We show the importance of each stage of the fitting process. Without stage 1 (optimizing for $\mathbf{z}$), our model performs worse on all metrics; this is also true for stage 2. Without stage 3, our model performs similarly, or slightly better, visually but suffers from a significant drop in ID reconstruction. The supplementary materials include visual results for each stage.

5.6. In the Wild Results

To demonstrate that our model is able to handle videos beyond those captured in controlled conditions, Fig. 8 includes results from webcam videos under different lighting conditions. While our method is already quite robust, we can easily add lighting variation using synthetic data in future work if we introduce a lighting model.

5.7. Runtime

After fitting a user's Avatar using the prior, we can generate the mesh attached Gaussian Avatar parameters, $\mathcal{A}$. Combined with the triangle face bindings and barycentric coordinates, this fully specifies an Avatar. No neural networks, including $\mathcal{D}$, are required for inference. A user's Avatar can be stored as an approximately 15MB file. Without any runtime optimizations, the complete inference pass, from Morphable Model parameters to the final rendered image runs at 70fps on an NVIDIA 4090 RTX GPU. The posing of the Gaussians can run at 67fps on a 3rd Gen Intel(R) Core(TM) i9-13900K CPU, suggesting improvements in Gaussian Splatting may allow real-time CPU inference.

6. Limitations and Future Work

While our model is able to achieve high-quality 360° rendering, it has some limitations. For some regions, such as the back of the head, the model produces synthetic-looking results. We would like to address this issue by looking into 2D image-based priors [26, 27] based on diffusion models [35]. To reduce artefacts introduced by overfitting to the monocular view, we used only flat RGB colour and did not model lighting, reducing our model's realism. In future, we may include a lighting model in our prior, enabled by a diverse set of lighting conditions in our synthetic data. As can be seen in our supplementary, our prior serves as a generative model with good interpretability. Given sufficient resources and a good camera/morphable model registration pipeline, we would like to use the findings of this work to train a similar generative prior using real data.

7. Conclusion

We have presented **GASP**, a novel method enabling 360°, high-quality Avatar synthesis from limited data. Our model builds a prior over Gaussian Avatar parameters to "fill in" missing regions. To bypass the issues associated with collecting a large-scale real dataset, such as the need for full coverage and imperfect annotation, we use synthetic data. Learned semantic Gaussian features and a three-stage fitting process enable us to cross the domain gap while fitting to real data to create realistic avatars. Our model outperforms the state-of-the-art in novel view and expression synthesis with Avatars trained from a single camera (e.g., a webcam or phone camera) using a short enrollment video or a single image while retaining the ability to animate and render in real-time.

References

[1] ShahRukh Athar et al. Bridging the gap: Studio-like avatar creation from a monocular phone capture. In *ECCV*, 2024. 2

[2] Shrisha Bharadwaj et al. Flare: Fast learning of animatable and relightable mesh avatars. *TOG*, 2023. 2

[3] Volker Blanz and Thomas Vetter. *A Morphable Model For The Synthesis Of 3D Faces*. Association for Computing Machinery, New York, NY, USA, 1 edition, 2023. 2

[4] Marcel C. Buehler, Gengyan Li, Erroll Wood, Leonhard Helminger, Xu Chen, Tanmay Shah, Daoye Wang, Stephan Garbin, Sergio Orts-Escolano, Otmar Hilliges, Dmitry Lagun, Jérémy Riviere, Paulo Gotardo, Thabo Beeler, Abhimitra Meka, and Kripasindhu Sarkar. Cafca: High-quality novel view synthesis of expressive faces from casual few-shot captures. In *SIGGRAPH Asia*. 2024. 2, 3, 8

[5] Marcel C Bühler, Kripasindhu Sarkar, Tanmay Shah, Gengyan Li, Daoye Wang, Leonhard Helminger, Sergio Orts-Escolano, Dmitry Lagun, Otmar Hilliges, Thabo Beeler, et al. Preface: A data-driven volumetric prior for few-shot ultra high-resolution face synthesis. In *ICCV*, pages 3402–3413, 2023. 2

[6] Chen Cao et al. Authentic volumetric avatars from a phone scan. *TOG*, 2022. 2

[7] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. In *SIGGRAPH*, pages 1–9, 2024. 2

[8] Cecilia Ding, Zheng ans Zhang, Zhihao Xia, Lars Jebe, Zhuowen Tu, and Xiuming Zhang. Diffusionrig: Learning personalized priors for facial appearance editing. In *CVPR*, 2023. 3, 6, 7

[9] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, pages 8649–8658, 2021. 2

[10] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Learning neural parametric head models. In *CVPR*, 2023. 4

[11] Simon Giebenhain, Tobias Kirschstein, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Npga: Neural parametric gaussian avatars. In *SIGGRAPH Asia*, 2024. 2

[12] Simon Giebenhain et al. Mononphm: Dynamic head reconstruction from monocular videos. In *CVPR*, 2024. 2

[13] Philip-William Grassal et al. Neural head avatars from monocular rgb videos. In *CVPR*, 2022. 2

[14] Yudong Guo, Keyu Chen, Sen Liang, Yongjin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *ICCV*, 2021. 2

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 5

[16] Charlie Hewitt, Fatemeh Saleh, Sadegh Aliakbarian, Lohit Petikam, Shideh Rezaeifar, Louis Florentin, Zafiirah Hosenie, Thomas J Cashman, Julien Valentin, Darren Cosker, and Tadas Baltrušaitis. Look ma, no markers: holistic performance capture without the hassle. *TOG*, 43(6), 2024. 2, 3, 5

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[18] Yang Hong et al. Headnerf: A real-time nerf-based parametric head model. In *CVPR*, 2022. 3, 7

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 3

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4), 2023. 2, 3, 4

[21] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *ECCV*, 2022. 3, 7

[22] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *TOG*, 37(4):163, 2018. 2

[23] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *TOG*, 42(4), 2023. 2, 5

[24] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *TOG*, 36(6):194:1–194:17, 2017. 2

[25] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022. 5

[26] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10083, 2024. 8

[27] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *ICCV*, pages 9264–9275, 2023. 8

[28] S. Ma, T. Simon, J. Saragih, D. Wang, Y. Li, F. La Torre, and Y. Sheikh. Pixel codec avatars. In *CVPR*, pages 64–73, Los Alamitos, CA, USA, 2021. IEEE Computer Society. 2

[29] Marko Mihajlovic et al. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. In *ECCV*, 2022. 2

[30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[31] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:

Representing scenes as neural radiance fields for view synthesis, 2020. 2

[32] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 4

[33] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *CVPR*, pages 20299–20309, 2024. 2, 3, 4, 6, 7

[34] Pramod Rao, Mallikarjun B R, Gereon Fox, Tim Weyrich, Bernd Bickel, Hanspeter Pfister, Wojciech Matusik, Ayush Tewari, Christian Theobalt, and Mohamed Elgharib. Vorf: Volumetric relightable faces. In *BMVC*. BMVA Press, 2022. 4

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 8

[36] Jack Saunders and Vinay Namboodiri. Dubbing for everyone: Data-efficient visual dubbing using neural rendering priors. *arxiv*, 2024. 2, 3

[37] Jack Saunders and Vinay P. Namboodiri. Read avatars: Realistic emotion-controllable audio driven avatars. In *arxiv*, 2023. 2

[38] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *CVPR*, 2024. 2, 3

[39] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6141–6150, 2020. 3

[40] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: real-time face capture and reenactment of rgb videos. *Commun. ACM*, 62(1):96–104, 2018. 2

[41] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: image synthesis using neural textures. *TOG*, 38(4), 2019. 2

[42] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. *ECCV 2020*, 2020. 2

[43] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *ICCV*, pages 3681–3691, 2021. 2, 3, 5

[44] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Xuhua Huang, Alexander Hypes, Taylor Koska, Steven Krenn, Stephen Lombardi, Xiaomin Luo, Kevyn McPhail, Laura Millerschoen, Michal Perdoch, Mark Pitts, Alexander Richard, Jason Saragih, Junko Saragih, Takaaki Shiratori, Tomas Simon, Matt Stewart, Autumn Trimble, Xinshuo Weng, David Whitewolf, Chenglei Wu, Shoou-I Yu, and Yaser Sheikh. Multiface: A dataset for neural face rendering. In *arXiv*, 2022. 2

[45] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *CVPR*, 2024. 2, 4, 6, 7

[46] Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *CVPR*, 2024. 2

[47] Yuelang Xu, Lizhen Wang, Zerong Zheng, Zhaoqi Su, and Yebin Liu. 3d gaussian parametric head model. In *ECCV*, 2024. 4

[48] Yuelang Xu et al. 3d gaussian parametric head model. In *ECCV*, 2024. 3

[49] Zhenhui Ye, Ziyue Jiang, Yi Ren, Jinglin Liu, Jinzheng He, and Zhou Zhao. Geneface: Generalized and high-fidelity audio-driven 3d talking face synthesis. *arXiv preprint arXiv:2301.13430*, 2023. 2

[50] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vision*, 129(11):3051–3068, 2021. 5

[51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4, 5

[52] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *CVPR*, 2023. 2

[53] Yufeng Zheng et al. Im avatar: Implicit morphable head avatars from videos. In *CVPR*, 2022. 2

[54] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *CVPR*, pages 4574–4584, 2022. 2

[55] Wojciech Zielonka et al. Instant volumetric head avatars. In *CVPR*, 2023. 2